REMARKS

Claims 1-3, 6-16, and 19-27 have been presented for examination in the above-identified U.S. Patent Application.

Claims 1-3, 6-16, and 19-27 have been rejected in Office Action dated March 14, 2003.

Claims 1, 6, 7, 9, 11, 14, 19-21, 23-25, and 27 have been amended by this Amendment A.

Claims 1-3, 6-16, and 19-27 are still in the Application and reconsideration of the application is hereby respectfully requested.

Referring to Office Action dated March 14, 2003, beginning in the middle of page 3 and extending to the end of page 5, Claims 1-3, 6-16, and 19-27 have been rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,598,560 issued in the name of Benson in view of U.S. Patent 5,859,963 issued in the name of O'Dowd et al., and in further view of U.S. Patent 5,933,641 issued in the name of Ma. The independent Claims of the Application (apparatus Claims 1 and 27 and method Claim 14) are directed to a program translation system that translates a program for a source device into a program for a target device. The claims include a computer-generated translation program. The translation program is then displayed on a user graphic interface device. The program elements for the source device program are displayed proximate to the related translation program elements for the target device. By displaying the source program elements proximate to the translation program elements, the user, using the graphic interface device, can change to translation elements to provide better conformity with the source program. This correction of the translation program can be performed prior to an actual execution of the translation program and can therefore expedite the generation of the translation program.

Turing to the Office Action, Benson discloses a translation system for translating a program for a source device into a translation program for a target device. Examiner, in the last two lines of page 2 indicates that "Benson doesn't explicitly disclose a backend including a graphic user interface." On Page 3, Examiner states that "O'Dowd does disclose this feature," i.e., "the input permitting a user to modify the translation elements based on a comparison with aligned source elements." A careful reading of the text (cf the O'Dowd Abstract), indicates that what is displayed on the screen, e.g., in Fig. 3 and Fig. 4, is the translation program arranged "in order according to the percentage of amounts of time spent in execution during of runs of the (sic translation) program". The apparatus then is "displaying the various selected source code lines in the order in which the lines are kept in the program along with the corresponding time spent by the program to execute machine code into which the source code has been compiled". (Quotations are from the Abstract) Note that one possible point of confusion is that the source code described in the O'Dowd reference is the type of code which generates the assembly language. In the present invention, the source program is derived from the source device as opposed to the target device and is not a description of the type of software program, i.e., source code, assembly language code, etc. Once this possible source of confusion is understood, then the display of O'Dowd reference includes what in the Application is referred to as the translation program (however, displayed in source code format to expedite user understanding) along with the translation program is the machine run time characteristics for each of the translation program (source code) elements. Consequently, what O'Dowd describes is not the improvement of the translation program derived from the source program, but rather is an optimization of the translation program without regard to the source program. This improvement/optimization is not what is described and claimed in the present Application. The present Application is directed at improving the relationship between the source program and the translation program.

The Ma reference has user interface 20 that is described as graphics or text. However, the Ma reference is directed to software program development and not to the translation of a program from one machine to another. In addition, the independent Claims 1, 14, and 27 have been amended to remove the assembly language limitation; this limitation being used for illustrative purposes in the Application.

In view of the discussion above, the Benson reference taken with the O'Dowd reference and/or the Ma reference describes in a reading most favorable to Examiner, a technique for optimizing a translation program. Nowhere is applicant able to find reference to a technique for expediting the translation of a program for a source device into a translation program. Consequently, the rejection of Claims 1, 14, and 27 under 35 U.S.C. 103(a) as being unpatentable over the Benson reference in view of the O'Dowd reference and in further view of the Ma is reference is respectfully traversed.

Because Claims 2, 3, 6-13, 15, 16, and 19-26 depend from Claims (1 and 14) now believed to be in condition for allowance, Claims 2, 3, 6-13, 15, 16, and 19-26 are believed to be in condition for allowance for the same reasons. The rejection of Claims 2, 3, 6-13, 15, 16, and 19-26 under 35 U.S.C. 103(a) as being unpatentable over the Benson reference in view of the O'Dowd reference in further view of the Ma reference is respectfully traversed.

In view of the foregoing discussion and the foregoing amendments, it is believed that Claims 1-3, 6-16, and 19-27 are now in condition for allowance of Claims 1-3, 6-16, and 19-27 is respectfully requested. Applicants hereby respectfully request a timely Notice of Allowance be issued in this Application.

Respectfully submitted,

William W. Holloway
Attorney for Applicants
Reg. No. 26,182

Texas Instruments Incorporated
PO Box 655474, MS 3999
Dallas, TX 75265
(281) 274-4064

Ms. Sharon

(972) 917-5348

VERSION WITH MARKINGS TO SHOW CHANGES MADE

Claim 1 has been amended as follows.

1. (Amended) A translation system for translating a source device ~~assembly language~~ program for a source device into a translation language program for a target device, the system comprising:

a front end for identifying source elements in a source device program; and

a back end for generating the translation program having translation elements corresponding to translation of the identified source device program elements, the backend including a graphic user interface, the graphic user interface displaying the identified source device program elements aligned with the corresponding translation elements, ~~display processor~~ the graphic user interface having an input unit, the input unit permitting a user to modify the translation elements based on comparison with the aligned source device program elements.

2. The system as recited in claim 1, wherein the source program is for a source device and the translation program is for a disparate target device.

3. The system as recited in claim 1, wherein the source program is a linear assembly file for a target device and the translation program is a scheduled assembly file for that device.

Claim 6 as has been amended as follows.

6. (Amended) The system as recited in claim 1, wherein the translation program is a context-dependent translation based on static analysis of the source program.

Claim 7 has been amended as follows.

7. (Amended) The system as recited in claim 1, wherein the back end further comprises:

a translator for performing a context-dependent translation, the translator comprising:

a translation machine description for mapping source opcodes to target opcodes;

a source machine description containing a description of source opcodes and source operands in a generic representation;

a target machine description containing a description of target opcodes and target operands in a generic representation; and

wherein the translator receives a source instruction from ~~said~~ the front end, utilizes the translation machine description and source machine description and target machine description to translate source elements into target elements.

8. The system as recited 7, wherein the proper target opcode is chosen from a group of potential target opcodes by comparing the target opcode and target operand with the source opcode and source operand.

Claim 9 has been amended as follows.

9. (Amended) The system as recited in claim 7, wherein two or more source opcodes can be combined to a single target opcode when there is a target opcode that represents the two or more source ~~code~~ opcodes.

10. The system as recited in claim 1, wherein the graphic user interface is a display processor.

Claim 11 has been amended as follows.

11. (Amended) The system as recited in claim 10, wherein the ~~graphical~~ graphic user interface displays at least a portion of the source elements in a source window, at least a portion of the translation elements in a translation window, and the source and translation windows are displayed side-by-side.

12. The system as recited in claim 11, wherein corresponding groups of elements of the source and translation programs are aligned in the source and translation windows.

13. The system as recited in claim 11, wherein at least one of the source and translation windows is operable to display a status icon for an element in the window.

Claim 14 has been amended as follows.

14. (Amended) A method for performing translation of ~~an assembly language~~ a source device program into ~~an assembly language~~ a translation program, the method comprising:

receiving a source ~~file~~ device program;

identifying source elements in the source device program;

generating a translation program having translation elements by performing a context-dependent translation of the source elements;

displaying the translation elements in a graphic user interface, the graphic user interface ~~for~~ receiving user inputs, the graphic user interface aligning the source ~~element~~ elements and the translation ~~element~~ elements, the aligned elements permitting a comparison of related source elements and translation elements; and

in response to user inputs, automatically regenerating selected translation elements based on the user inputs.

15. The method as recited in claim 14, wherein the source program is for a source device and the translation program is for a disparate target device.

16. The method as recited in claim 14, wherein the source program is a linear assembly program for a target device and the translation program is a scheduled assembly program for the target device.

Claim 19 has been amended as follows.

19. (Amended) The method as recited in claim 14, further comprising:

performing static analysis of the source elements in the source device program; and

performing context-dependent translation of the source elements based on the static analysis.

Claim 20 has been amended as follows.

20. (Amended) The method as recited in claim 14, wherein the step of generating a translation program further comprises:

converting an opcode of a source ~~machine~~ device program to an opcode of a translation machine program by comparing the source ~~opcode~~ opcodes to possible translation opcodes;

converting the operand of the source opcode by comparing an operand of the source opcode in a generic expression with a generic expression for a translation operand; and

combining the translation opcode and the translation operand to form a translation.

Claim 21 has been amended as follows.

21. (Amended) The method as recited in claim 20, wherein the step of converting an opcode of the source ~~file~~ program further comprises choosing a translation opcode from a group of potential translation opcodes by comparing the translation opcode and translation operand with the related source opcode and source operand.

22. The method as recited in claim 20, wherein the step of converting the source opcode further comprises the step of combining two or more source opcodes into a single translation opcode when there is a translation opcode that represents the two or more source opcodes.

Claim 23 has been amended as follows.

23. (Amended) The method as recited in claim 14, wherein the ~~user interface is a graphical~~ graphic user interface includes a display processor.

Claim 24 has been amended as follows.

24. (Amended) The method as recited in claim 23, further comprising:

displaying the source elements in a source window;

displaying the translation elements in a translation window; and

displaying the source and translation windows side-by-side in the ~~graphical user interface~~ display processor.

Claim 25 has been amended as follows.

25. The method as recited in claim 24, further comprising aligning corresponding groups of elements of the source and translation ~~program~~ programs in the source and translation windows.

26. The method as recited in claim 24, further comprising displaying a status icon for an element in at least one of the source and translation windows.

Claim 27 has been amended as follows.

27. (Amended) A translation system for translating a source device program into a translation program for a target device, the system comprising:

a computer capable of executing a program, and

an interactive program for translating code for ~~a first processor~~ the source device into code for ~~a second processor~~ the target processor and capable of being executed on ~~said~~ the computer; and

a graphics interface system displaying source program elements proximate to corresponding translation program elements, the graphics interface system permitting ~~the~~ a comparison of corresponding source program elements and translation program elements, the graphic interface unit having a user input device, the user input device permitting correction of the translation program elements by a user ~~based on~~ as a result of the comparison.